

Künstliche Intelligenz in der Ingenieurgeodäsie

Alexander Reiterer

Zusammenfassung

In der modernen Ingenieurgeodäsie werden zunehmend komplexe Sensorsysteme bzw. -verbunde gefordert. Diese erfordern auch die Entwicklung neuer Ansätze für die Steuerung, die Auswertung sowie die Interpretation der erfassten Daten. Im Bereich der künstlichen Intelligenz (KI) besteht eine Reihe von Techniken, die für eine effiziente Lösung solcher Probleme eingesetzt werden können. Dieser Artikel gibt einen Überblick über die wichtigsten Techniken aus diesem Bereich und ihre möglichen Einsatzgebiete in der Ingenieurgeodäsie.

Summary

Complex sensor respectively multi sensor systems are increasingly needed in modern Engineering Geodesy. They require the development of new approaches for controlling, analyzing and the interpretation of captured data. A variety of techniques which can be used for an efficient solution of such problems exists in the area of artificial intelligence (AI). We report on the most important techniques of AI and their operational area in Engineering Geodesy.

1 Einleitung

Die Entwicklung der künstlichen Intelligenz (KI) ist mit vielen Missverständnissen, Wunschvorstellungen und falschen Eindrücken verbunden. Im Jahre 1956 haben führende Computerwissenschaftler, Psychologen, Linguisten und Philosophen am Dartmouth College im Zuge eines Kongresses den Begriff »Artificial Intelligence« (künstliche Intelligenz) geschaffen. Hintergrund war es, ein präzises Modell des menschlichen Denkens zu erstellen – die Schaffung eines »Elektronenhirns« sollte Computern die Fähigkeiten des biologischen Vorbildes geben. Der große Optimismus der Pionierjahre wird unter anderem durch das »General Problem Solve« Projekt, bei dem ein universelles Computerprogramm zur Lösung unterschiedlicher Planungsaufgaben geschaffen werden sollte, widerspiegelt. Das Projekt scheiterte an der unüberschaubaren Anzahl von möglichen Plänen.

Von Anbeginn bestand eines der Hauptprobleme der KI in der schwierigen Definition ihrer selbst. Was kann als »intelligent« bezeichnet werden, was als »künstlich«. Das Verhalten einer Maschine wurde damals als intelligent angesehen, wenn diese den sog. Turing-Test bestand, d. h. wenn das Verhalten von der entsprechenden Leistung eines Menschen nicht zu unterscheiden war (Görz et al. 2000). Der Turing-Test hat bis heute seine Berechtigung und findet nach wie vor Anwendung. Die »Intelligenz« einer Maschine oder eines Programms ausschließlich an die nicht mögliche Unterscheidbarkeit vom Menschen zu

koppeln, scheint aber zu kurz gegriffen. Intelligenz wird heute vielmehr mit Kreativität und selbstständiger Lösungsfindung in Verbindung gebracht.

Im alltäglichen Sprachgebrauch werden häufig Begriffe wie Denkvermögen, Auffassungsgabe, Rationalität, Logik, Urteilsvermögen und Kreativität verwendet, um die geistigen Fähigkeiten eines Menschen zu kennzeichnen. Diese bleiben als Konstrukt jedoch so lange inhaltsleer, bis sie nicht messbar gemacht und von anderen abgegrenzt werden (Zimbardo 1995). Die Schwierigkeit der Definition von Intelligenz wird wohl durch die Aussage »Intelligenz ist das, was Intelligenztests messen« am besten verdeutlicht. Gardner (2002) definiert Intelligenz wie folgt: »Intelligenz ist das biopsychologische Potential zur Verarbeitung von Informationen, das in einem kulturellen Umfeld aktiviert werden kann, um Probleme zu lösen oder geistige oder materielle Güter zu schaffen, die in einer Kultur hohe Wertschätzung genießen.«

Von »künstlich« wird heute einheitlich dann gesprochen, wenn etwas nicht durch einen natürlichen Prozess entstanden ist, also durch Menschen oder Maschinen erzeugt wurde.

Computer haben heute ein außergewöhnliches Leistungspotential erreicht – sie rechnen genauer als der Mensch, sind bei logischen Spielen meist besser und sind vor allem bei sämtlichen Rechenaufgaben schneller. Probleme haben sie aber bis heute bei der Bewältigung von unklaren, ihnen unbekanntem Situationen. Nach der oben angeführten Definition können sie also klar als »nicht-intelligent« eingestuft werden. Wie kann nun die KI dazu beitragen, unseren heutigen »Rechenmaschinen« Intelligenz zu verleihen?

Das Forschungsgebiet der KI kann heute in zwei große Gruppen unterteilt werden. Die erste Gruppe beschäftigt sich mit der möglichst exakten Nachbildung der menschlichen Denkweise. Bei der Entwicklung neuer Techniken stützt man sich auf Erkenntnisse aus dem Bereich der Psychologie und der Neurologie. In diese Kategorie fallen z. B. künstliche neuronale Netze. Zusammengefasst wurden solche Techniken unter dem Begriff des *Soft Computing*. Die zweite Gruppe beschäftigt sich mit der Entwicklung von Systemen, deren intelligentes Verhalten auf Methoden beruht, die sich von den von Menschen benutzten Methoden unterscheiden. Diese Vorgehensweise stützt sich auf eine typisch ingenieurtechnische Herangehensweise. Beispiel einer solchen Entwicklung sind wissensbasierte Systeme. Bei diesen Techniken spricht man sehr oft von *Hard Computing*.

In Zukunft wird künstliche Intelligenz weniger in einzelnen Programmen stecken als vielmehr in ihrem Verbund. Als Beispiel seien hier sog. Smart Devices angeführt. Sie stellen winzige Prozessoren und Sensoren dar,

die in diversen Endgeräten fest integriert sind und untereinander »telematisch kommunizieren« (Verbmobil 2006).

Dieser Artikel soll einen allgemeinen Überblick über die Techniken der KI und spezielle Anwendungen in der Ingenieurgeodäsie geben. Es ist hier jedoch nicht möglich, eine vollständige Abhandlung darzulegen; Schwerpunkt soll auf die für die Ingenieurgeodäsie und Messtechnik wichtigsten Methoden gelegt werden. Für einen umfangreichen Überblick sei auf die Literatur verwiesen (Görz et al. 2000).

2 Techniken der KI

Heute können eine Unzahl von unterschiedlichen Techniken und Methoden aus dem Bereich der KI in der Ingenieurgeodäsie und Messtechnik eingesetzt werden. Wir möchten uns hier auf vier der bekanntesten Gruppen beschränken: künstliche neuronale Netze, wissensbasierte Systeme, evolutionäre Algorithmen (bzw. genetische Algorithmen) und (Multi)-Agenten-Systeme.

2.1 Künstliche neuronale Netze

Unter einem künstlichen neuronalen Netz (KNN) versteht man die Kombination einer nichtlinearen Funktion mit einem Lern-Algorithmus. Der Lern-Algorithmus dient dazu, aus vorhandenen Eingangs- und gewünschten Ausgangswerten alle Parameter der Funktion zu bestimmen (Zell 1994). Ein KNN besteht aus einer Vielzahl von gleichen Einzelteilen, den sog. künstlichen Neuronen und stellt somit eine Realisierung des konnektionistischen Paradigmas dar.

Jedes KNN besteht in seiner minimalen Form aus einer Eingabeschicht (input layer) und einer Ausgabeschicht (output layer) – meist ist noch eine sog. Zwischenschicht (hidden layer) vorhanden. Die Zwischenschicht determiniert die Fähigkeit des Netzes zur Verallgemeinerung. Die Anzahl der Neuronen in der Zwischenschicht muss groß genug sein, um die gestellte Aufgabe zu erfüllen, sie muss aber gleichzeitig klein genug sein, um eine sinnvolle Verallgemeinerung durch das Netz zu ermöglichen (Zell 1994). Die Anzahl der Neuronen in der Zwischenschicht bestimmt die Verknüpfungen innerhalb des Netzes und damit die Komplexität des Netzes¹. Die Neuronen der Eingabeschicht sind mit den Neuronen der Ausgabeschicht über die Zwischenschicht mit gewichteten Verknüpfungen verbunden (siehe Abb. 1). Diese Verknüpfung kann auf unterschiedlichste Art und Weise ausgeführt sein: voll-

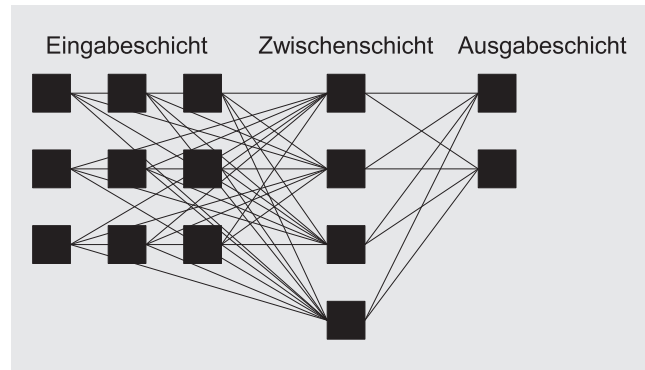


Abb. 1: Vollständig verbundenes KNN (3 × 3 – 1 × 4 – 1 × 2)

ständige Verknüpfung oder Teilverknüpfungen zwischen gewissen Neuronen, vorwärtsgerichtete Verknüpfung oder beidseitig gerichtete Verknüpfung.

Die Berechnung der Gewichte dieser Verknüpfungen erfolgt dann durch einen geeigneten Lernalgorithmus. Für dieses »Lernen« benötigt man eine ausreichend große Anzahl von Trainingsdaten und einen geeigneten Trainingsalgorithmus.

Der bekannteste und am weitesten verbreitete Trainingsalgorithmus ist der *Backpropagation-Algorithmus*. Er arbeitet als iteratives Verfahren und ermittelt die Konfiguration der Gewichte im Netz, indem er die Fehler-summe über alle Trainingsmuster minimiert. Weitere bekannte Trainingsalgorithmen sind das Kohonen oder das Monte-Carlo-Verfahren. Auf die vollständige Beschreibung der Trainingsalgorithmen soll hier verzichtet werden; es sei auf die einschlägige Literatur verwiesen (Görz et al. 2000, Zell 1994).

In der Praxis besteht oft das Problem, dass die Eingangsdaten redundante Informationen enthalten und diese auch nicht durch geeignete Codierung aufgehoben werden können. Um diesem Problem entgegenzuwirken, sollte das KNN die »Filterung« der redundanten Informationen vornehmen können. Erreicht werden kann dies durch ein geeignetes Netz-Design, sodass die Zwischenschicht weniger Neuronen als die Eingabeschicht besitzt.

Zusammenfassend kann gesagt werden, dass sich KNN vor allem für Probleme eignen, bei denen man eine ausreichend große Menge von (Trainings)-Daten zur Verfügung hat und keine genaue Kenntnis über den algorithmischen Zusammenhang zwischen diesen und den gewünschten Ausgangsdaten hat – der Zusammenhang zwischen Eingabe- und Ausgabedaten soll gelernt werden.

Typische Aufgaben für künstliche neuronale Netze (in der Ingenieurgeodäsie und Messtechnik) sind die Klassifikation und die Mustererkennung.

2.2 Wissensbasierte Systeme

Wissensbasierte Systeme (WBS) unterscheiden sich in ihrer Architektur grundsätzlich von Programmen, welche in einer konventionellen Programmiersprache (Java, C, C++ u. v. a. m.) implementiert wurden. Zu den Hauptkom-

¹ In einem Netz mit einer Zwischenschicht ist die Anzahl der Verknüpfungen proportional zu der Anzahl der Neuronen in der Zwischenschicht.

ponenten eines wissensbasierten Systems (Abb. 2) zählen: Wissensbasis, Inferenzkomponente, Benutzerschnittstelle, Erklärungskomponente und Wissenserwerbskomponente.

Die Wissensbasis enthält das Problemwissen in deklarativer Form. Dieses Wissen wird durch die Inferenzkomponente verarbeitet, wobei neues Wissen in

Form von Fakten abgeleitet wird. Die Benutzerschnittstelle muss zumindest in einer minimalen Form vorhanden sein. Zusätzlich kann jedes wissensbasierte System über eine Wissenserwerbskomponente, welche die Akquisition neuen Wissens erleichtern soll, und über eine Erklärungskomponente, welche dem Benutzer Auskunft über die konkrete Lösungsfindung geben soll, verfügen (Gottlob et al. 1990, Stefik 1998).

Vorteil eines wissensbasierten gegenüber einem herkömmlichen Ansatz ist vor allem die klare Trennung von Problemwissen und Wissensverarbeitung, d.h. dem Programmcode an sich. Erweiterungen und Modifikationen der Wissensbasis sind dadurch wesentlich leichter möglich. Ein weiterer Vorteil ist, dass Expertenwissen häufig in Form von Regeln vorhanden ist. Dieses Wissen kann ohne Konvertierung in die Wissensbasis eingebracht werden. Für die praktische Implementierung wissensbasierter Systeme existieren verschiedene Ansätze (prozedurale Methoden, objektorientierte Methoden, logikbasierte Methoden u.a.) und verschiedene Softwarehilfsmittel, z.B. Clips, Ilog, u.a. (Clips 2004, Ilog 2004).

In der Praxis werden heute meist Mischformen für die Implementierung eingesetzt; die wohl am meisten verbreitete Methode ist die regelbasierte/objektorientierte Kombination. Ein solches System besteht im Wesentlichen aus zwei Teilen: dem so genannten Arbeitsspeicher (working memory – WM) und der Regelbasis (rule memory). Der Arbeitsspeicher ist eine Sammlung von einzelnen Typen sowie deren Instanzen (vergleichbare Formen sind `record` in Pascal oder `struct` in C). Die eigentlichen Elemente des Arbeitsspeichers (working memory elements) sind dann konkrete Instanzierungen. Der zweite Teil eines regelbasierten Systems ist die Regelbasis. Eine Regel ist grundsätzlich in zwei Teile unterteilt, nämlich in den Bedingungsteil (lefthand side – LHS) und den Aktionsteil (righthand side – RHS). Im Bedingungsteil stehen die so genannten Vorbedingungen, welche für das Aus-

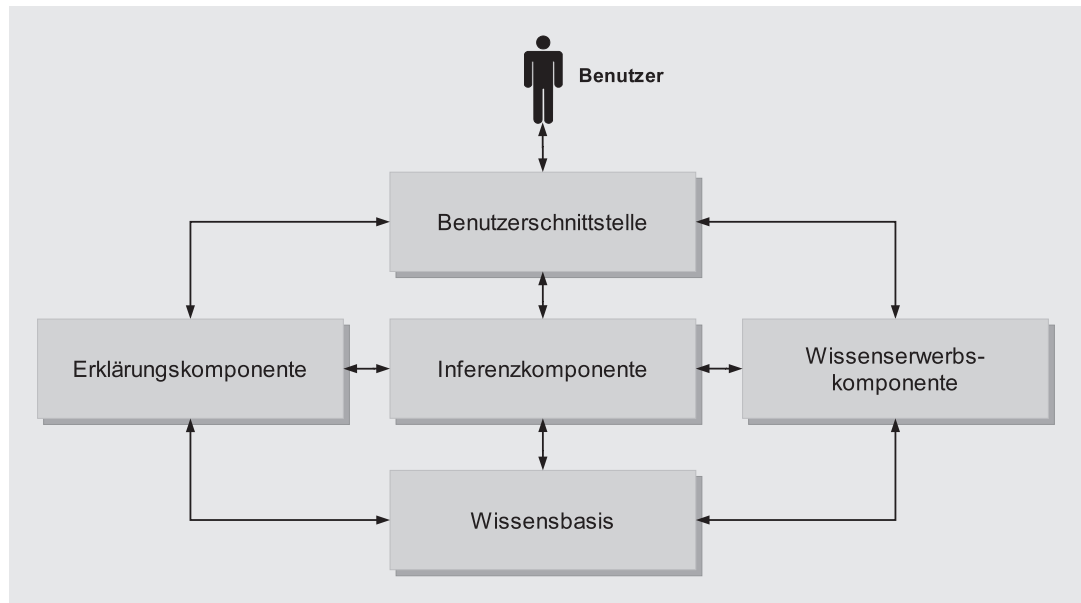


Abb. 2: Aufbau eines wissensbasierten Systems (Gottlob et al. 1990)

führen der Regel erfüllt sein müssen. Erst wenn alle Vorbedingungen erfüllt sind, wird der Aktionsteil der Regel ausführbar. Beispiel einer solchen Regel:

```

(defrule NAME
  (Vorbedingung 1) (Vorbedingung 2) (...)
  (Vorbedingung n)
=>
  (Aktion))
  
```

Als enge Verwandte von wissensbasierten Systemen können sog. Fuzzy-Systeme gesehen werden. Sie bestehen im Wesentlichen aus denselben Komponenten wie wissensbasierte Systeme, verfügen aber zusätzlich über die Fähigkeit, unscharfes Wissen zu verarbeiten (Kruse 1994, Michels 2002). Im praktischen Einsatz können wissensbasierte Systeme immer auch als Fuzzy-Systeme konzipiert und realisiert werden.

Wissensbasierte Systeme lassen sich in der Ingenieurgeodäsie unter anderem sehr gut für diverse Konfigurationsaufgaben und für sog. Decision-Support Systems einsetzen. Beispiele hierfür sind Systeme für die Konfiguration und Steuerung von Datenverarbeitungsschritten und Frühwarnsysteme.

2.3 Evolutionäre Algorithmen (genetische Algorithmen)

Evolutionäre Algorithmen orientieren sich direkt am Evolutionsmodell der Natur. Darwins Hypothese war, dass die Selektion als treibende Kraft für die Evolution und für Weiterentwicklung bzw. Differenzierung der Individuen verantwortlich ist. Aber erst die molekularbiologische Sichtweise hat uns die Details für diesen Prozess gezeigt. Demnach codiert die DNA (Desoxyribonucleid Acid) sämtliche Informationen, die den Phänotyp (das

Individuum) charakterisieren, angefangen von seiner morphologischen Gestalt bis hin zu seinem Verhalten. Das Lebensalter wird nur indirekt durch die DNA bestimmt, indem gewisse Verfahren (z.B. die Selektion) das Überleben der einzelnen Individuen beeinflussen.

Wichtigster Punkt für eine algorithmische Beschreibung des evolutionären Modells ist die Vermehrung; mit ihrer Hilfe kann Mutation und Crossover (die Variation gegenüber den Elternindividuen) umgesetzt werden. Vereinfacht gesagt kann also jeder Algorithmus, der eine Variation durchführt und das »Bessere« überleben lässt, als eine Nachbildung der Evolution gesehen werden. Dies kann aber nur als eine übergeordnete Strategie gesehen werden – zu vage und zu weit gefasst ist diese Definition.

2.4 (Multi)-Agenten-Systeme

Agenten-Systeme stellen eines der bedeutendsten Paradigmen der neueren Informatik dar. Das Konzept beruht auf der Idee, ein System aus weitgehend selbstständigen »intelligenten« (Software-)Einheiten als Stellvertreter realer Objekte (z.B. Benutzer, Informationsquellen u. a.) zu schaffen. Eine einheitliche Definition ist wie bei vielen anderen Techniken der KI (siehe oben) schwierig – fundamental für Agenten ist jedoch der Begriff der Autonomie. Eine mögliche Definition stammt von Wooldridge (1999): »An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.

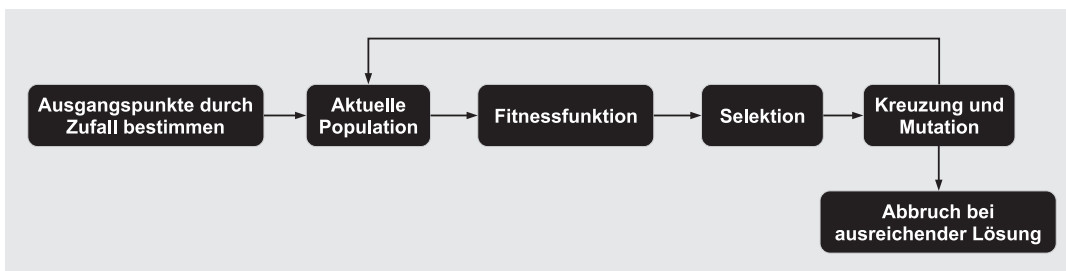


Abb. 3: Funktionsweise eines genetischen Algorithmus (Polheim et al. 2000)

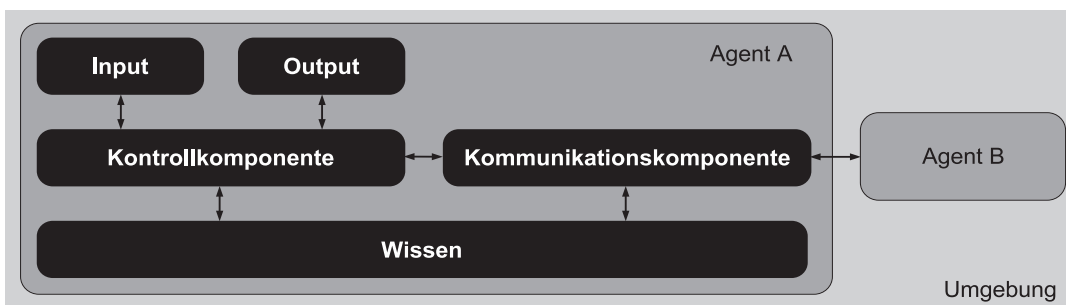


Abb. 4: Vereinfachter Aufbau eines Multi-Agenten-Systems

Ein evolutionärer Algorithmus sollte ein stochastisches Optimierungsverfahren sein², bei dem die Selektion und/oder Variation zufällige Einflüsse darstellen.

Heute gibt es im Wesentlichen zwei Entwicklungsgruppen für evolutionäre Algorithmen: die Evolutions-Strategen und die Vertreter der genetischen Algorithmen (Holland 1975). Der Ablauf eines genetischen Algorithmus ist in Abb. 3 dargestellt. Dabei ist zu beachten, dass als Voraussetzung für die Umsetzung eines Problems als evolutionärer bzw. genetischer Algorithmus die Möglichkeit der Codierung als Tupel gegeben sein muss.

Genetische Algorithmen eignen sich in der Ingenieurgeodäsie für sämtliche Probleme der Optimierung. Beispielhaft sei hier auf die Optimierung der Punkterfassungsmethoden und des Matchings bei bildgebenden Sensoren hingewiesen (Ebner 1998). Eine detaillierte Abhandlung des Funktionsprinzips evolutionärer und genetischer Algorithmen findet sich in Gerdes et al. (2004) und Polheim et al. (2000).

Agents are able to act without the intervention of humans or other systems: they have control both over their internal state, and over their behaviour.«

Die Abgrenzung hin zu wissensbasierten Systemen gestaltet sich nach dieser Definition als schwierig. Der wesentliche Unterschied liegt darin, dass wissensbasierte Systeme und im Speziellen Expertensysteme in ihrer klassischen Form als einzige Interaktionsquelle den Benutzer besitzen. Des Weiteren besitzen sie keinerlei selbstständige Handlungsmöglichkeiten. Im praktischen Einsatz gibt es (besonders in den Ingenieurwissenschaften) eine Reihe von wissensbasierten Systemen, die Agenten sehr ähneln. Für die praktische Umsetzung gibt es (wie auch bei WBS, siehe Abschnitt 2.2) eine Reihe unterschiedlicher Ansätze und Modelle. Zu den wichtigsten zählen dabei logikbasierte, reaktive (reactive) und BDI-Architekturen. Logikbasierte Ansätze zählen zu den traditionellen Techniken in der KI – die Umgebung und das Verhalten werden dabei in symbolischer Notation realisiert. Bei der reaktiven Architektur entsteht das Verhalten aus der Interaktion mit einfachen Verhaltensmustern. Die BDI-Methoden haben ihren Ursprung in der Philosophie; die Entscheidungsfindung erfolgt dabei zweistu-

2 Im Gegensatz zur klassischen Optimierung finden evolutionäre Algorithmen das Maximum und nicht das Minimum.

fig: (1) Deliberation – Auswahl der Wünsche (desires) aus den Möglichkeiten, die uns aufgrund der Annahmen über die Umwelt (beliefs) zur Verfügung stehen; (2) Means-ends reasoning – Auswahl der Absichten (intentions), die verfolgt werden sollen. Die Kommunikation der Agenten untereinander erfolgt über Kooperationsprotokolle. Dabei werden komplexe Aufgaben in kleine Teile unterteilt und zugewiesen. Der abstrahierte Aufbau eines Multi-Agenten-Systems ist in Abb. 4 dargestellt.

Im praktischen Einsatz findet man Multi-Agenten-Systeme vor allem dort, wo Systeme aus einer Vielzahl von einzelnen komplexen Komponenten bestehen (z. B. Multi-Sensor-Systeme). Jede einzelne Komponente ist dabei für sich gesehen ein eigenes, abgeschlossenes System, das in eine Umwelt (Umgebung) eingebettet ist, mit dieser in Wechselwirkung steht und daher als Agent betrachtet werden kann.

3 Praktischer Einsatz

In den letzten Jahren wurden im Bereich der Ingenieurgeodäsie vermehrt Techniken aus dem Bereich der KI eingesetzt. Der Schwerpunkt lag dabei bei wissensbasierten Systemen und künstlichen neuronalen Netzen. Erwähnenswert sind vor allem die wissensbasierte Interpretation von Brückendeformationen (Samba-Endbericht 1995), die genetische Optimierung von Punkterfassungsalgorithmen (Ebner 1998), die neuronale Modellierung von Bauwerks-Deformationsprozessen (Heine 1999, Miima 2002), die Analyse von Verschiebungsdaten im Tunnelbau (Chmelina 2002), die wissensbasierte Steuerung von bildgebenden Sensorsystemen (Reiterer et al. 2003, Reiterer 2004), der Einsatz von Fuzzy-Methoden zur Detektion konsistenter Punktbeziehung (Haberler 2004) und die wissensbasierte Steuerung von Multi-Sensor-Systemen (Thienelt et al. 2005a und 2005b).

Es ist völlig klar, dass die meisten Probleme der Ingenieurgeodäsie (gilt auch für die meisten anderen Ingenieurwissenschaften) auch mit traditionellen Methoden der Softwareentwicklung bewältigbar wären/sind. Im Bereich des Softwareengineering hat sich jedoch gezeigt, dass durch die Umstrukturierung des Entwicklungsprozesses und den Einsatz neuer Techniken die Qualität der Produkte wesentlich erhöht werden kann (Sommerville 2001).

Um einen besseren Eindruck von möglichen Einsatzgebieten solcher Techniken zu bekommen, seien nachfolgend einige Beispiele aus der Praxis angeführt. Für eine ausführliche Beschreibung sei auf die jeweilige Literatur verwiesen.

3.1 Steuerung von Konfigurationsaufgaben

In vielen Bereichen der Ingenieurgeodäsie kommen sog. Konfigurationsaufgaben vor; dazu zählen z. B. die Erstellung

von optimalen Abläufen für hochkomplexe Messaufgaben, oder die Erstellung von Algorithmenfolgen für eine optimale Lösung oder Konfigurationsaufgaben im Bereich der Netzplanung. Unter Konfiguration versteht man ganz allgemein die Aufgabe, aus einer fixen Menge von Komponenten gewisse Elemente so auszuwählen und anzuordnen, dass sie eine gegebene Spezifikation erfüllen. Jede Konfigurationsaufgabe besteht im Wesentlichen aus einer Menge von Komponenten (z. B. Algorithmen), einer Beschreibung der gewünschten Konfiguration und optimalen Kriterien zur Optimierung der Lösung. Ideal für die Umsetzung solcher Aufgaben sind wissensbasierte Systeme. Das Grundprinzip einer solchen Realisierung beruht in der Formulierung sämtlicher Zusammenhänge als Regeln. Dieses Regelwerk wählt dann auf Grundlage gewisser Fakten (diese können online erhoben/berechnet werden oder dem System bereits aus vorangegangenen Berechnungen vorliegen) eine geeignete Konfiguration aus. Beispielhaft sei hier auf die Konfiguration von Algorithmen für bildgebende Messsysteme in Reiterer (2004) verwiesen. Die Aufgabe besteht dabei darin, aus einem Grundstock von Algorithmen (Bildaufbereitung und Punkterfassung) die geeignetsten auszuwählen, sie in die geeignetste Reihenfolge zu bringen und die Parameter, welche für ihre Ausführung notwendig sind, zu wählen. Endziel ist es, geeignete Punkte für die Objektrekonstruktion bzw. die Deformationsanalyse zu erfassen. Die Grundlage für diesen Konfigurationsprozess bilden Merkmale, die online aus den erfassten Bildern extrahiert werden (statistische Größen). Anschließend an diese Berechnung werden die Werte in qualitative Werte umgerechnet und gemeinsam mit den Ursprungswerten in geeigneter Form gespeichert. Bei dieser Umrechnung werden numerische, metrisch skalierte Werte in symbolische, nominal skalierte Werte konvertiert. Dies ermöglicht es, in Regeln abstrakte Werte heranzuziehen. Die Umrechnung ist ein Spezialfall einer »Fuzzifizierung« der metrischen Werte, die den Grad der Zugehörigkeit eines metrischen Wertes zu jedem der symbolischen Werte festlegt.

Nach durchgeführter Konfiguration wird dem Benutzer die Möglichkeit gegeben, kritische Algorithmen (z. B. informationsreduzierende Filter) aus dieser Auswahlliste zu entfernen oder die komplette Systementscheidung zu verwerfen. Anschließend werden für sämtliche Schritte die notwendigen Parameter bestimmt und in einem letzten Schritt die Reihenfolge ihrer Anwendung festgelegt. Die getroffene Auswahl wird in eine sog. Taskliste geschrieben und einer Systemkontrollkomponente übergeben, welche die einzelnen Algorithmen ausführt und die Ergebnisse in geeigneter Form speichert. Der Aufbau des wissensbasierten Bildaufbereitungssystems ist in abstrahierter Form in Abb. 5 dargestellt.

Der Prozessablauf ist für Bildaufbereitung und Punkterfassung weitgehend identisch. Nach durchgeführter Bildaufbereitung wird jedoch eine Neuberechnung der Bildmerkmale durchgeführt. Einen ähnlichen Ansatz wählte auch Mason (1994) für die Erstellung einer ge-

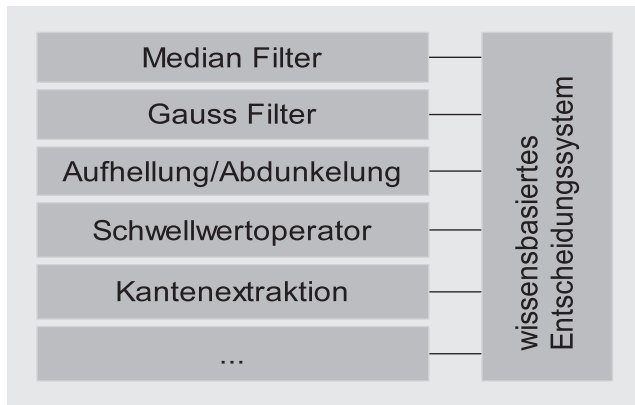


Abb. 5: Wissensbasiertes Bildaufbereitungssystem

eigneten Netzkonfiguration für photogrammetrische Aufnahmen. Dabei bettete er das gesamte Wissen über eine ideale Netzkonfiguration in ein wissensbasiertes System ein.

Solche regelbasierten Ansätze haben jedoch den Nachteil, dass sie meist domänen-spezifisches Wissen und Kontroll-Wissen miteinander verweben und dadurch schwierig zu warten sind. Daher benutzt man in den letzten Jahren vermehrt modellbasierte Methoden – sie behandeln Konfigurationsprobleme als generische Aufgaben (Gottlob et al. 1990, Puppe 1991, Stefik 1998).

Angemerkt sei, dass solche Konfigurationsaufgaben auch in ein Optimierungsproblem übergeführt werden und dann mit Hilfe von genetischen Algorithmen gelöst werden könnten. Ein weiterer möglicher Ansatz könnte auf einem Multi-Agenten-System beruhen. Dabei wird jeder Algorithmus als eigener Agent repräsentiert; eine optimale Lösung ergibt sich dann durch die Kommunikation der einzelnen Agenten untereinander und durch das global formulierte Ziel.

3.2 Selbstlernende Systeme für »Black-Box-Anwendungen«

Die oben angeführten Beispiele zeigen sehr deutlich, dass für die regelbasierte Implementierung das Wissen in expliziter Form vorhanden sein muss³. Dies impliziert jedoch, dass man den Zusammenhang zwischen den Eingangsdaten und den gewünschten Ausgangsdaten eines Systems genauestens kennt. In vielen Fällen ist dieser Zusammenhang jedoch nicht bekannt oder auf Grund der Komplexität des Systems nicht mit vergleichbar einfachen Mitteln wie Regeln modellierbar (als Beispiel sei hier das Deformationsverhalten komplexer Objekte angeführt). In solchen Fällen werden meist lernende Systeme eingesetzt, die es erlauben, den Zusammenhang zwischen

³ Bei wissensbasierten Systemen ist auch möglich, die Regeln selbstständig lernen zu lassen. Diese Methoden resultieren jedoch in unstrukturierten Wissensbasen; der große Vorteil solcher Systeme (Wissen in strukturierter Form vorliegen zu haben) geht dadurch verloren.

vorhandenen Eingangsdaten und Ausgangsdaten zu bestimmen. Mit Hilfe dieses Zusammenhanges kann später das Ausgangsmuster auf Grundlage eines neuen Eingangsmusters »vorhergesagt« werden. Im praktischen Einsatz finden sich für solche Problemstellungen meist künstliche neuronale Netze oder neuronale Fuzzy-Systeme. In diesem Zusammenhang kann man von »Black-Box-Anwendungen« sprechen, da das Wissen nicht »frei verfügbar« im System vorhanden ist, sondern fest in den Gesamtprozess verwoben ist (im Fall von künstlichen neuronalen Netzen steckt das Wissen in den erlernten Gewichten der Verbindungen zwischen den einzelnen Neuronen – siehe Abschnitt 2.1). Als praktisches Beispiel

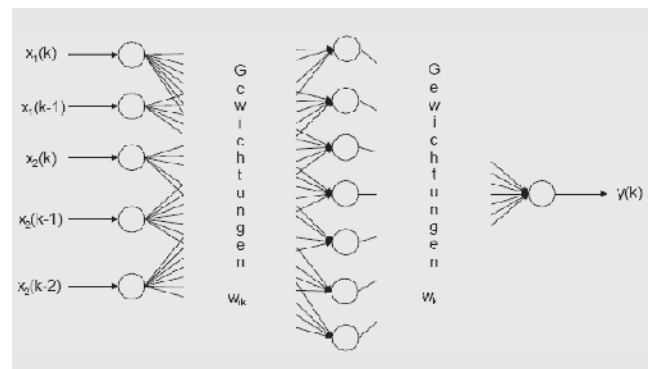


Abb. 6: Struktur eines künstlichen neuronalen Netzes zur Modellierung eines Deformationsprozesses mit zwei Einflussgrößen (Heine 1999)

sei hier die Modellierung von Bauwerksdeformationen mit Hilfe eines neuronalen Netzes erläutert (Heine 1999, Miima 2002).

Das Deformationsverhalten von Bauwerken kann meist schon mit sehr einfachen künstlichen neuronalen Netzen modelliert werden. Als Eingangsgrößen dienen dabei im einfachsten Fall die erfassten Messgrößen und als Ausgangsgrößen die Deformationen des Objektes. Das künstliche neuronale Netz verknüpft nun diese beiden Daten-gruppen miteinander. Bei der Wahl der Trainingsdaten ist zu beachten, dass sie die auftretenden Deformationen möglichst gut und umfassend repräsentieren sollten – nur so ist eine spätere zuverlässige Vorhersage möglich (dazu ist es im Bereich der Deformationsmessungen meist nötig, Daten über einen langen Zeitraum zu sammeln; Miima (2002) berichtet in diesem Zusammenhang von Messungen, die über drei Jahre hindurch durchgeführt wurden, um das komplexe dynamische Verhalten einer Brücke zu modellieren). Ein möglicher Aufbau eines künstlichen neuronalen Netzes für die Modellierung einer Bauwerksdeformation ist in Abb. 6 dargestellt.

Nachdem die Trainingsdaten zur Verfügung stehen und damit die Form der Eingabe- und Ausgabeschicht weitgehend festgelegt ist, geht es um das Design des Gesamtnetzes. Für diesen Prozess gibt es in der Praxis jedoch keine einheitlichen Regeln (Zell 1994). Meist ist es empfehlenswert, nicht ein vollständig verbundenes künstliches neuronales Netz zu verwenden, sondern die Ver-

bindungen auf Grundlage der geforderten Eigenschaften festzulegen (Bishop 1995). Im praktischen Einsatz hat sich gezeigt, dass der größte Erfolg dadurch erreicht wird, wenn von einem relativ simplen Netz ausgegangen und dieses dann je nach Bedarf verfeinert wird.

4 Schlussbemerkung

Dieser Artikel hat einen groben Überblick über Techniken aus dem Bereich der künstlichen Intelligenz gegeben, welche heute bei Problemen der Ingenieurgeodäsie und Messtechnik eingesetzt werden können. Es zeigt sich sehr deutlich, dass mit diesen Verfahren in den verschiedensten Bereichen sehr gute Ergebnisse erzielt werden können. Vor allem in der Steuerung, Konfiguration und Optimierung haben Beispiele aus der Praxis das Potenzial dieser Techniken klar aufgezeigt. Für eine breite Akzeptanz dieser Techniken scheint es jedoch sehr wichtig zu sein, dass anwenderfreundliche Tools zur Verfügung stehen, die eine schnelle und unkomplizierte Implementierung solcher Systeme garantieren.

Trotz aller Fortschritte im Bereich der KI sollte mit dem Begriff »Intelligenz« im Zusammenhang mit Programm- und Sensorsystemen vermehrt vorsichtig umgegangen werden. Leider stößt man nur allzu oft auf sog. »intelligente Systeme«, die bei einer genaueren Betrachtung dieser Definition (trotz aller Schwierigkeit, diese zu formulieren) nicht standhalten können – allzu schnell wird von intelligenter Steuerung, intelligenten Messsystemen oder von einer intelligenten Auswertung gesprochen; eine spezifischere Benennung wäre wünschenswert.

Abschließend kann gesagt werden, dass die KI ein immenses Potential für die Software- und Hardwareentwicklung darstellt und in vielen Ingenieurdisziplinen erfolgreich eingesetzt werden kann und auch bereits eingesetzt wird. In der Ingenieurgeodäsie gibt es zahlreiche Problemstellungen, die mit Hilfe dieser Techniken wesentlich effizienter gelöst werden könnten – einem erfolgreichen Einsatz steht somit nichts mehr im Wege.

Literatur

- Bishop, Chr.: Neural networks for pattern recognition. Clarendon Press, 1995.
- Chmelina, K.: Wissensbasierte Analyse geodätischer Verschiebungsmessungen. Dissertation, TU-Wien, 2002.
- CLIPS-Webseite: <http://www.ghg.net/clips/CLIPS.html>, 2006.
- Ebner, M.: On the Evolution of Interest Operators using Genetic Programming. First European Workshop on Genetic Programming, The University of Birmingham, 1998.
- Förstner, W.: Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung, DGK, Vol. C, Nr. 370, München, 1991.
- Gardner, H.: Intelligenzen – Die Vielfalt des menschlichen Geistes. Klett-Cotta Verlag, Stuttgart, 2002.

- Gerdes, I., Klawonn, F., Kruse, R.: Evolutionäre Algorithmen. Vieweg Verlag, Wiesbaden, 2004
- Gottlob, G., Frühwirth, Th., Horn, W. (Hrsg.): Expertensysteme. Springer Verlag, Wien, 1990.
- Görz, G., Rollinger, C.-R., Schneeberger, J.: Handbuch der künstlichen Intelligenz. Oldenbourg Wissenschaftsverlag, München, 2000.
- Haberler, M.: Einsatz von Fuzzy-Methoden zur Detektion konsistenter Punktbeziehung. Dissertation, TU-Wien, 2004.
- Heine, K.: Beschreibung von Deformationsprozessen durch Volterra- und Fuzzy-Modelle sowie neuronale Netze. Dissertation, Technische Universität Cottbus, 1999.
- Holland, J.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Michigan, 1975.
- ILOG – White Paper: <http://www.ilog.com>, 2006.
- Kruse, R., Gebhardt, J., Klawonn, F.: Foundations of Fuzzy Systems. Wiley Verlag, Chichester, United Kingdom, 1994.
- Mason, S.: Expert System-Based Design of Photogrammetric Networks. Dissertation, ETH-Zürich, 1994.
- Michels, K., Klawonn, F., Kruse, R., Nürnberger: Fuzzy Regelung. Springer-Verlag, Heidelberg, 2002.
- Miima, J.-B.: Artificial Neural Networks and Fuzzy Logic Techniques for the Reconstruction of Structural Deformations. Dissertation, TU-Braunschweig, 2002.
- Polheim, H.: Evolutionäre Algorithmen. Springer Verlag, Wien/New York, 2000.
- Puppe, F.: Einführung in Expertensysteme. Springer Verlag, Berlin, 1991.
- Reiterer, A.: Knowledge-Based Decision System for an On-line Videotheodolite-based Multisensor System. Dissertation, TU-Wien, 2004.
- Reiterer, A., Kahmen, H., Egly, U., Eiter, Th.: Knowledge-Based Image Preprocessing for a Theodolite Measurement System. In: Optical 3-D Measurement Techniques VI, Vol. I, Zürich, 2003.
- SAMBA-Endbericht – Projekt: Wissensbasiertes System »Messtechnik im Bauwesen«. Technische Universität Hannover, 1995.
- Somerville, I.: Software Engineering. Addison-Wesley Verlag, 2001.
- Stefik, M.: Introduction to Knowledge Systems. 2. Aufl., Kaufmann Verlag, San Francisco, 1998.
- Thienelt, M., Eichhorn A., Reiterer A.: Konzept eines wissensbasierten KALMAN-Filters für die Fußgängerortung (WiKaF). Österreichische Zeitschrift für Vermessung und Geoinformation (VGI), 2005.
- Thienelt, M., Eichhorn A., Reiterer A.: WiKaF – A Knowledge-Based Kalman-Filter for Pedestrian Positioning. In: Location based Services Telectography – Proceedings of the Symposium, 2005.
- Verbmobil-Webseite: <http://verbmobil.dfki.de>, 2006.
- Wooldridge, M.: Intelligent Agents. In: Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.
- Wooldridge, M.: Intelligent agents – Theory and Practice. In: The Knowledge Engineering Review, 1995.
- Zell, A.: Simulation Neuronaler Netze. 1. Aufl., Addison-Wesley Verlag, 1994.
- Zimbardo, Ph.: Psychologie. Springer Verlag, Wien/New York, 1995.

Anschrift des Autors

Univ. Ass. Dipl.-Ing. Dr. techn. Alexander Reiterer
 Forschungsgruppe Ingenieurgeodäsie
 Institut für Geodäsie und Geophysik
 Technische Universität Wien
 Gußhausstraße 27–29
 1040 Wien, Österreich
alexander.reiterer@tuwien.ac.at